Multimedia Systems
Lecture – 26

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

Arithmetic Coding

- Unlike the methods discussed previously, arithmetic coding overcomes the requirement of every symbol to be coded independently.
- In other words, coding every symbol was represented individually by a code, or a group was represented. Thus, a whole number of bits were required to encode a symbol (or symbol group).
- Arithmetic coding overcomes this constraint by mapping an entire message to a real number between zero and one. This real number representing the entire message is coded as a binary number.
- Arithmetic coding, thus, encodes a message entirely without assigning a fixed binary code to each symbol and, thereby, tends to produce better compression ratios.

- The coding process uses a one-dimensional table of probabilities instead of a tree structure.
- Given an alphabet of n symbols, there are an infinite number of messages that are possible. Each message is mapped to a unique real number in the interval [0,1).
- The interval contains an infinite amount of real numbers, so it must be possible to code any message uniquely to one number in the interval.
- The interval is first set at [0,1) for the first symbol, and then partitioned according to the symbol probabilities.

- The algorithm can be outlined as follows:
  - Divide the interval [0,1) into n segments corresponding to the n symbols; the segment of each symbol has a length proportional to its probability. Each segment i has an upper bound U and lower bound L corresponding to the start of the segment and the end of the segment (U L Pi).
  - Choose the segment that corresponds to the first symbol in the message string. This is the new current interval with its computed new upper and lower bounds.
  - Divide the new current interval again into n new segments with length proportional to the symbols probabilities and compute the new intervals accordingly.
  - From these new segments, choose the one corresponding to the next symbol in the message.
  - Continue Steps 3 and 4 until the whole message is coded.
  - Represent the segment’s value by a binary fraction in the final interval.

- Example -1

Symbol statistics

Symbol	Probability	Cumulative Distribution	low	high
A	0.4	0.4	0	0.4
B	0.4	0.8	0.4	0.8
L	0.2	1.0	0.8	1.0

Encoding Sequence BALL

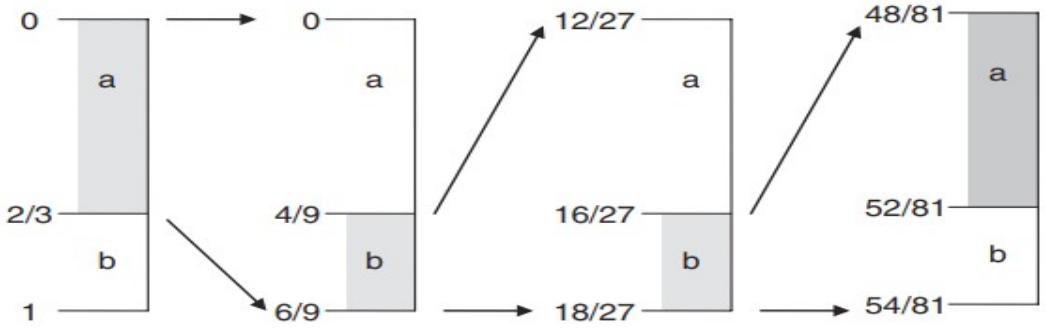
Encode ‘B’: low=0+0.4*1=0.4 high=0+0.8*1=0.8

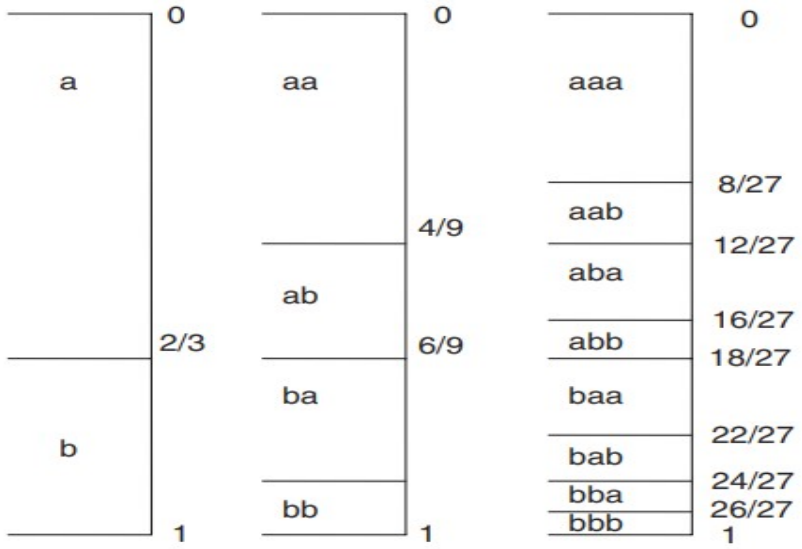
Encode ‘A’: low=0.4+(0)*(0.4)=0.4 high=0.4+(0.4)(0.4)=0.56

Encode ‘L’: low=0.4+(0.8)*(0.16)=0.528 high=0.4+(1.0)(0.16)=0.56

Encode ‘L’: low=0.528+(0.8)*(0.032)=0.5536 high=0.528+(1.0)(0.032)=0.56

- Example -2 Two symbols a, b having probabilities as follows P(a) 2/3, P(b) 1/3. To encode the message of length 4 “abba”





Adaptive Arithmetic Coding

- Like adaptive Huffman coding, adaptive arithmetic coding is also possible.
- The difference between the two is that there is no need to keep a tree for the codewords. The only information that needs to be synchronized is the frequency of occurrence of the symbols.
- Unlike the previous example, the statistics table will be updated as symbols are encoded. The symbols are also updated when symbols are decoded.

- Initial table

symbol	frequency	probability	low	high
A	4	0.4	0	0.4
B	4	0.4	0.4	0.8
L	2	0.2	0.8	1

Encode ‘B’: low=0+0.4*1=0.4 high=0+0.8*1=0.8

Table after ‘B’ is encoded

symbol	frequency	probability	low	high
A	5	4/11	0	4/11
B	4	5/11	4/11	9/11
L	2	2/11	9/11	1

Encode ‘A’: low=0.4+(0)*(0.4)=0.4 high=0.4+(0.4)(4/11)=6/11

Table after ‘A’ is encoded

symbol	frequency	probability	low	high
A	5	5/12	0	5/12
B	5	5/12	5/12	10/12
L	2	2/12	10/12	1

Encode ‘L’: low=0.4+(8/55)*(10/12)=86/165 high=0.4+(1.0)(8/55)=0.56

Table after ‘L’ is encoded

symbol	frequency	probability	low	high
A	5	5/13	0	5/13
B	5	5/13	5/13	10/13
L	2	3/13	10/13	1

Encode ‘L’: low=(86/165)+(10/13)*(44/1815)=0.53986 high=(86/165)+(1)(44/1815)=0.54545

Assignment

- A,B, and C, P(A) = 0.5, P(B) = 0.25 and P(C) =0.25, Find out the arithmetic code for BACA.